ICTNET at Web Track 2010 Ad-hoc Task

Xu Chen[12], Zeying Peng[12], Jianguo Wang[12], Xiaoming Yu[1], Yue Liu[1], Hongbo Xu[1], Xueqi Cheng[1]

1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190

2. Graduate School of Chinese Academy of Sciences, Beijing, 100190

Abstract

In this paper, our team – "ICTNET", participated in the ad-hoc task of Web Track of TREC 2010. The full Category A dataset was used. The sliding window BM25 model was extended from last year's method, combining with link analysis to rank the final results. All the methods having been tried in our experiments are delineated, and the evaluation results from the organizers of Web Track are presented thereafter.

1. Introduction

Ad-hoc task in TREC investigates the performance of systems that search a static set of documents using previously-unseen topics. The goal of the task is to return a ranking of the documents in the collection in the order of decreasing probability of relevance [1]. The probability of relevance of a document is considered independently of other documents that appear before it in the result list. Ad-hoc task focuses on document relevance. The results will be assessed mainly by ERR, precision, NDCG as well as MAP.

This year, ClueWeb09 dataset [2] is continued to use. In our system, the full Category A which consists of around 500 million English pages is used. The amount of data is a real challenge for our data processing capability.

In section 2, data preparation steps are described. Section 3 gives the description of the retrieval methods for different fields that was used in experiments. Our ranking methods combined various scores from different fields are given in section4. Section 5 gives the conclusion.

2. Data Preparation

1) System

We used Firtex [3], an open source, high performance distributed search platform developed by our lab in recent years. Firtex was deployed over 10 servers, each of which has 8 CPU cores, 16GB memory and 1.4TB hard disk. A Python search frontend for servers and a Python client were also developed. While performing search, the client sends the query to all ten servers, and servers scan through their part of data, rank their local results and then return them to the client in an asynchronous manner.

2) Content Extraction

The content from all pages were extracted before the indexing was performed. At the beginning, some useless part of the pages were tried to be expurgated, for example, advertisements and navigation columns. But it was found that there are great varieties among the dataset. On one hand, it was hard to define uselessness; on the other hand, it is almost impossible to find a general expurgation method for all pages, many of which fail to obey the HTML standard. Thus, we simply extracted title, links, anchors and all visible text as content. Also the keywords in the metadata parts are also extracted for later analysis.

Report Documentation Page

1. REPORT DATE **NOV 2010**	2. REPORT TYPE	3. DATES COVERED **00-00-2010 to 00-00-2010**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
ICTNET at Web Track 2010 Ad-hoc Task	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Chinese Academy of Sciences,Institute of Computing Technology,Beijing, 100190,**	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at the Nineteenth Text REtrieval Conference (TREC 2010) held in Gaithersburg, Maryland on 16-19 November 2010. The conference was co-sponsored by the National Institute of Standards and Technology (NIST), the Defense Advanced Research Projects Agency (DARPA), and the Advanced Research and Development Activity (ARDA).

14. ABSTRACT
In this paper, our team ? ?ICTNET?, participated in the ad-hoc task of Web Track of TREC 2010. The full Category A dataset was used. The sliding window BM25 model was extended from last year?s method, combining with link analysis to rank the final results. All the methods having been tried in our experiments are delineated, and the evaluation results from the organizers of Web Track are presented thereafter.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**4**	

3) Indexing

After titles, keywords and contents had been extracted, these fields were indexed into inverse index by Firtex. It took about eight hours to index all Category A data onto ten machines, and each of them had 120GB of data. When a query is submitted, it takes approximately 30 seconds to fetch all results from the servers by the client.

4) Link Analysis

After extracting links and anchors from all pages, link analysis and anchor texts extraction were performed. These were computed in a map-reduce way. Anchor texts were merged and duplicates of every URL were eliminated because a large amount of anchor texts are meaningless for ad-hoc retrieval, such as "Home", "About", etc.

Having the in-links and out-links, PageRank [4] values were calculated for all pages. The classic PageRank algorithm was run until convergence was reached. It was noticed that many pages in the dataset do not have in-link while some others have thousands. This leads to great imbalance among the PageRank values.

3. Retrieval Model

1) Sliding Window BM25

The most commonly used retrieval model is BM25 [5], based on the probabilistic retrieval framework developed in the 1970s and 1980s. This method has been the baseline for retrieval methods. The basic equation is:

$$BM25(Q,D) = \sum_{q \in Q} \frac{f(q,D) \bullet (k_1 + 1)}{f(q,D) + k_1 \bullet (1 - b + b \bullet \frac{|D|}{avgdl})} \bullet \log \frac{N - df(q) + 0.5}{df(q) + 0.5},$$

where Q is query containing term q; $f(q,D)$ is the term frequency of q in document D; $|D|$ is the length of D; $avgdl$ is the average length of total N documents; $df(q)$ is the document frequency of q; $k1$ and b are parameters.

One problem of BM25 model is that term proximity is not taken into account. In last year's ad-hoc task, we proposed the minimum window BM25 [6], in which the basic idea is that if all terms of a query appear in a smaller area, the document is more likely to be relevant, and should receive a higher rank. In this method, documents are ranked as following:

$$BM25_{window}(Q, Doc_{content}) = \frac{f_w \times boost \times \sum_{q \in Q} BM25(q)}{\log(w+1)}, w \le MAXWINDOW.$$

where Q is a query containing terms q; $DOC_{content}$ is the content of a document; BM25(q) is the BM25 score of query term q; w is the minimum window size that containing all query terms; $MAXWINDOW$ is the upper limit of w; f_w is the frequent of different minimum window in $DOC_{content}$; $boost$ is the weight of Q. A fixed threshold $MAXWINDOW$ was defined to govern how long the distance between query words appeared in a document could be tolerated. If one document contains more windows which cover the query words, and if the windows' size is relatively smaller, this document could get a higher score.

The above method alleviates the problem of the original BM25, but still has a drawback: the threshold *MAXWINDOW* is pre-defined, lacking of flexibility. If the length of one query is too long, this method is not well suited here for a fixed window size may not be big enough to contain all query words. With this consideration, this year an improvement was made to let the *MAXWINDOW* vary with the length of a query. In fact the *MAXWINDOW* was calculated as:

$$MAXWINDOW = length(Q) + k \times length(Q),$$

where *length(Q)* denotes the number of words of one query, and *k* is a parameter which can be tuned. When performing search, the retriever system moves the window along every document and accumulates the frequencies of different windows whose sizes do not exceed the *MAXWINDOW*.

This method was used in the content field, because a large amount of pages have a long content.

2) Search with Titles

Usually the title of a web page is a short and compact description of the content. If the title closely relates to the query, this page may be well relevant. Also search the title would not harm the performance too much for the titles' shortness.In the title field, the BM25 model was not employed, but a simple Boolean retriever. The Boolean connective "OR" was used, otherwise for some long queries, there may be too few documents found.

3) Search with Keywords

Here keywords are extracted from meta-data parts of a page, which usually have the form as:

<**META content**="Template 59 Business/General Business Templates General Web Templates Sharp Business Template Sharp Simple design with warm colors neat navigation" **name**="Keywords">.

This part is used to do search engine optimization, so it often gives fine abstract of the page. These keywords are generally longer than title, so they were considered to better reveal the intention of the content.In this part, the original BM25 model was used, because keywords are not as long as contents, and there is no need to use sliding window.

4. Ranking

The different scores from different fields, the PageRank value and spamming were combined together to get the final ranker. The combined ranking function is in the following form:

$$S(Q,D) = [S_{fields}(Q,D) + \beta_1 S_{PR}(D)] \square I(S_{spam}(D) > 50),$$

where $S_{PR}(D)$ is the log smoothed PageRank value of the document D; β_1 is a weight for PageRank value; $S_{spam}(D)$ denotes the spamming weight in the Waterloo Spam Rankings for the ClueWeb09 Dataset [7] and I is indicator function.

The $S_{fields}(Q,D)$ in the right hand side of the above equation is the score of the combination of the three fields. The content, title and keyword fields were combined by simply using an "AND" connective in the Firtex retriever. Actually, the retriever would sum up the scores calculated for each fields with the methods mentioned in the previous section. Different weights for each field could also be assigned, thus the score is computed like this:

$$S_{fields}(Q, D) = \omega_1 \square BM25_{window}(Q, DOC_{content}) + \omega_2 \square Sim_{bool}(Q, DOC_{title}) + \omega_3 \square BM25(Q, DOC_{keyword}),$$

where $BM25_{window}$, Sim_{bool} and $BM25$ denote the sliding window BM25, the simple Boolean model and the original BM25 model respectively.

5. Results

We submitted three runs. The first run, ICTNETAD10R1, used sliding window BM25 to model the content of document, meaning that the parameters ω_1 and ω_2 were set to zero in S_{fields}. The second run, ICTNETAD10R2, added title field and keyword field as described above. PageRank values were not combined in the first two runs. The third run submitted, ICTNETAD10R3, used sliding windows BM25 in the content field as the first run, and re-ranked the results by combining their PageRank values. All runs were anti-spammed by the Waterloo Spam Rankings.All runs were evaluated by NDCG@20, ERR@20, P@10, P@20 and MAP.The results are listed in Table 1.

Run	Relevant Retrieved	ERR@20	NDCG@20	P@10	P@20	MAP
ICTNETAD10R1	1677	0.1015	0.1799	0.3750	0.3736	0.0932
ICTNETAD10R2	1143	0.0964	0.1473	0.3028	0.2944	0.0552
ICTNETAD10R3	1153	0.0662	0.1089	0.2139	0.2097	0.0374

Table 1: Evaluation Results

6. Conclusion

In this paper, our methods used in ad-hoc task of the TREC 2010 Web Track were described. We demonstrated our steps of system and data preparation, the way we modeled the document and query, and our ranking method.It is noticed from the results in Table 1 and Table 2 that, the first run using the simplest method is the best while the third run with PageRank the worst. Adding extra fields in the second didn't help. Maybe because simple is beauty.

Acknowledgements

References

[1] http://plg.uwaterloo.ca/~trecweb/2010.html

[2] http://boston.lti.cs.cmu.edu/Data/clueweb09/

[3] http://sourceforge.net/projects/firtex/

[4] Page, L.; Brin, S.; Motwani, R. and Winograd, T., The PageRank Citation Ranking: Bringing Order to the Web, Technical report, Stanford Digital Library Technologies Project, 1998

[5] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In Proceedings of the Third Text REtrieval Conference, Gaithersburg, USA, November 1994

[6] F. Guan, X. Yu, Z. Peng, H. Xu, Y. Liu and L. Song. ICTNET at Web Track 2009 Ad-hoc Task. In Proceedings of the Eighteenth Text REtrieval Conference, Gaithersburg, Maryland, November 17-20, 2009

[7] http://plg.uwaterloo.ca/~gvcormac/clueweb09spam/